

REMARKS

I. Introduction

In response to the Office Action dated March 1, 2007, claims 1 and 5 have been amended, and claims 17-18 have been added. Claims 1-18 remain in the application. Re-examination and re-consideration of the application, as amended, is requested.

II. Specification Objections

The disclosure was objected to for use of acronyms without first including a description in plain text. Applicants have amended the specification to include a description in plain text of the acronyms recited in the Office Action. Accordingly, Applicants submit that the objections are now moot.

The disclosure was objected to for the use of the trademarks Netscape Navigator™ and Microsoft Internet Explorer™ without capitalization wherever they appear. Applicants note that the trademarks Netscape Navigator™ and Microsoft Internet Explorer™ only appear once each in the specification in paragraph [0023] on page 7 of the originally filed specification. In that paragraph, the terms are already in full capitalization and easily distinguishable from the remaining text. Accordingly, Applicants submit that efforts have been made to prevent their use in any manner which might adversely affect their validity as trademarks. In view of the above, Applicants respectfully request withdrawal of the objections.

III. Examiner Interview Summary

Record is made of a two telephone interviews in connection with the present patent application as follows:

(1) On May 15, 2007 between Applicants' attorney Jason S. Feldmar, Inventor John G. Beltran, Inventor Phillip D. Beymer, Inventor David Stroud, Examiner Omar R. Ali, and Examiner James Myhre; and

(2) On May 17, 2007 between Applicants' attorney Jason S. Feldmar, Inventor John G. Beltran, Inventor David Stroud, Examiner Omar R. Ali, and Examiner James Myhre.

Applicants appreciate the time and effort taken by the Examiner in speaking with and discussing the present application.

During both interviews, the current claims and the above amendments were discussed (in a proposed form) in view of the prior art. More specifically, the independent claims were discussed in view of the Ramchandani reference (U.S. 6,486,893).

Agreement was not reached in that Examiners and Applicants were unable to resolve or establish the differences between the cited reference and the proposed claims.

IV. Prior Art Rejections

In paragraphs (3)-(4) of the Office Action, claims 1-12 were rejected under 35 U.S.C. §102 as being anticipated by Ramchandani et al., U.S. Patent 6,486,893 (Ramchandani). In paragraphs (5)-(6) of the Office Action, claims 13-16 were rejected under 35 U.S.C. §103(a) as being obvious in view of Ramchandani.

Specifically, the following independent claims were rejected as follows:

As to claim 1, Ramchandani discloses a method and system for a property browser, which displays sub-objects in an Active X, control comprising:

- a. receiving a reference to an object instance (standard Active X control) having a dynamic property that is supplied at runtime for the object instance on a per instance basis and is not stored with the object (column 5, lines 11-15);
- b. retrieving a reference to a property source instance associated with the object, wherein the property source instance (type library) maintains knowledge of the dynamic property (Col 6, lines 5-16);
- c. providing the reference to the object and the reference to the property source to a control (property browser) which is configured to: (i) retrieve the dynamic property from the property source and (ii) display the property in a user interface (column 6, lines 5-39).

As to claim 5, Ramchandani discloses a method and system for a property browser, which displays sub-objects in an Active X, control comprising:

- a. an object instance having a dynamic property that is supplied at runtime for the object instance on a per-instance basis and is not stored with the object (column 5, lines 11-15);
- b. a property source instance associated with the object, wherein the property source instance maintains knowledge of the dynamic property (column 6, line 5-15);
- c. a host (container) configured to: (i) retrieve a reference to the object instance; (ii) retrieve a reference to the property source; (iii) provide the reference to the object and the reference to the property source to a control which is configured to: (1) retrieve the dynamic property from the property source; and (2) display the property in a user interface (column 6, lines 5-16).

As to claim 9, Ramchandani discloses a method and system for a property browser, which displays sub-objects in an Active X, control comprising:

- a. receiving a first object having a first property, wherein the first object provides a custom ActiveX control that defines a first user interface for displaying and editing the first property (column 5, lines 61-67 to column 6, lines 7-16);
- b. creating a list of one or more object properties to be displayed, wherein the list includes the first property (column 6, lines 7-16);
- c. instantiating the custom ActiveX control (column 5, lines 61-67 to column 6, lines 1-6);

d. displaying the object properties in the list, wherein the display of the first property comprises the first user interface defined by the instantiated custom ActiveX control, wherein the property may be edited through the first user interface (column 6, lines 7-16).

As to claim 13, Ramchandani discloses a method and system for a property browser, which displays sub-objects in an Active X, further comprising:

a. one or more objects, wherein each object has one or more object properties (column 6, lines 5-16);

b. a property inspector (property browser) configured to: (i) interrogate the one or more objects to discover one or more object properties to be displayed; (ii) create a list of the one or more object properties to be displayed; and (iii) instantiate and host one or more property editors (column 6, lines 5-46);

c. Ramchandani does not explicitly disclose (i) one of the property editors comprises a custom Active X control specified by one of the objects and (ii) the custom Active X control defines a custom graphical user interface for displaying and editing one of the object properties. However, Ramchandani does disclose multiple Active X control objects may be placed onto the form and the property browser is able to display indicators to expand sub-object properties for purposes of manipulating the values of the child properties of the sub-object property (column 6, lines 2-46). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made that additional Active X controls specified by individual objects could be used for displaying and editing object properties. One would have been motivated to incorporate custom Active X controls into property editors to provide custom properties that are specific to each control.

Applicants traverse the above rejections. As indicated in the telephonic interviews with the Examiners, the claims are detailed in their structure and limitations. There are essentially two sets of claims. The first set includes independent claims 1 and 5 (and new claim 17). The second set includes claims 9 and 13. Each set will be addressed separately herein.

Claims 1 and 5 (and New Claim 17)

These independent claims provide the ability to create and display a dynamic property on a per object basis. As explicitly set forth in the claims, the dynamic property is created at runtime for an object instance on a per-instance basis and is not stored with the object. Referring to claim 1, the first limitation provides for retrieving a reference to an object instance. This element provides an instance of an object/class for which the dynamic property will be created.

The second limitation retrieves a reference to a property source instance. Applicants note that the limitation provides that it is an instance of the property source (i.e., it is an instance of an object). Further the reference to this property source instance is obtained. The second limitation also provides that the reference is retrieved from “an association between the object and the property source instance”. Thus, rather than retrieving a reference from a random location, it is

retrieved from an entity identified as an “association” in the claims. Such an association can take a variety of forms (e.g., a mapping, a linked list, etc.). In addition, the second element provides that the property source instance creates and supplies the dynamic property and an initial value for the dynamic property for/to the object instance. Accordingly, when examining this claim limitation in combination with the first limitation, the claims explicitly provide that a property source instance creates (at run time) and supplies (at run time) a new property and an initial value for the new property to/for the object instance. Thus, rather than the object instance creating a value for one of its own properties at runtime (e.g., a name for the object instance), this claim element explicitly provides that a property itself AND a value for that property is created for one object (i.e., the object instance) by another object (i.e., the property source instance).

The third limitation provides the two references (i.e., the reference to the object instance and the reference to the property source instance) to a control. The third limitation further provides that the control is configured to retrieve the dynamic property from the property source instance and display the dynamic property in a user interface.

As can be seen by the above limitations, there are specific and detailed claim limitations that provide a precise structure and capability. More specifically, a property source instance creates a property (and a value for that property) for an object instance. Such a creation is performed dynamically during runtime.

The Office Action rejects the above claims based on Ramchandani. The Ramchandani reference and the present application were also discussed in detail during the two interviews. During the interviews, an attempt was made to map the various claim elements to the entities of Ramchandani. Applicants note that difficulty arose while attempting such mapping because it is impossible to map the elements of the present claims to Ramchandani. Ramchandani relates to a property browser window that is capable of displaying properties of sub-objects in an ActiveX control. However, the detailed and explicit claim limitations of the present invention are neither taught nor disclosed by Ramchandani.

The Examiners asserted (during the interview) that Ramchandani taught the creation of the property since the user has the ability to edit the properties of the ActiveX control including the deletion of a property. The Examiners asserted that since Ramchandani can delete a property, the addition of a property would be obvious. Applicants again direct the attention to the individual

claim limitations and respectfully request that the Patent Office describe where Ramchandani teaches each of the claim limitations both alone and in combination. In this regard, assuming that the claimed object instance is a sub object of the ActiveX control, it is impossible for Ramchandani to teach the remaining claim elements. Firstly, the claim limitations provide for an association between the object instance and the property source instance (i.e., a separate instance of a property source object/class). Such an association does not exist in Ramchandani. Secondly, a reference to a property source instance is retrieved from the association. Examining such a limitation, Ramchandani does not provide for a retrieval of a reference to a property source instance from any association. Further, even if such a reference to a property source instance exists in Ramchandani, the capability for such a property source instance to create and supply the dynamic property (and an initial value for such a dynamic property) at run time is non-existent in Ramchandani.

Applicants note that Ramchandani's FIG. 10 illustrates a screen shot of a property browser (on the right hand side of the figure) editing the properties of a CWSlide ActiveX object (in the left hand side of the object). The user has the capability to edit the values of the properties using the property browser (see col. 8, lines 24-54). Such properties could potentially be deleted by the user as well. However, the present invention does not rely on the user to perform any dynamic property creation or deletion. Instead, a separate object, the property source instance, creates and supplies the dynamic property for the object instance at runtime. As claimed, the property source is an instance of an object, and a reference to such a property source instance is retrieved from an association. The existence and use of such a property source instance is entirely and completely lacking from Ramchandani. Again, no association exists in Ramchandani and no property source instance is used to create a dynamic property.

In addition to the above, Applicants submit that the capability for a user to delete a property does not teach, disclose, or suggest, the capability to add a property. In this regard, Ramchandani's properties all exist in a type library at compile time. All of the properties are displayed in the property browser. The user can then elect to delete a property. However, there is no capability described or that suggests the addition of a property. To add a property would be contrary to Ramchandani's teaching and use of the type library since the type library cannot be accessed or added to during runtime. Instead, Ramchandani merely removes the display of the property from the property window. The present invention has an object (the property source instance) that

creates an actual property and a value for that property - all of which are for another object (the object instance). Such a dynamic property creation is wholly and completely lacking from Ramchandani.

Lastly, the ability for a control to use the two references and retrieve and display the dynamic property is lacking from Ramchandani. In this regard, Ramchandani does not teach, disclose, or suggest the ability to both create a property and then display such a newly created property using the unique combination of object instances, association, and references as claimed.

Claims 9 and 13

These independent claims are directed towards the display of a property in a property list. More specifically, once an object with a property has been retrieved, that same object provides an ActiveX control that defines a user interface for displaying and editing the property. A list of properties is created. The ActiveX control is used to display a user interface for one of those properties in the list. Thus, multiple ActiveX controls are used to display individual properties in a property list.

Applicants submit that Ramchandani clearly fails to teach, disclose, or suggest such unique attributes as set forth in the present claim limitations. Rather than utilizing a separate ActiveX control to display individual properties, Ramchandani describes the use of a property browser window to edit properties of an ActiveX control. In this regard, Ramchandani describes something wholly and completely different from the present invention. Namely, rather than teaching an ActiveX control that defines a user interface for editing a property, Ramchandani describes a property of an ActiveX control.

Applicants refer again to Fig. 10 of Ramchandani. The CWSlide object in the left panel window of Fig. 10 is an ActiveX control slider. The property window on the right of Fig. 10 lists the various properties of the CWSlide ActiveX control. For example, the caption "Temperature" in the property window (the word "Caption" is highlighted in the property window indicating that it is being edited by the user) is reflected by the text that is vertical in the left panel window. Such an example cannot teach and cannot be compared to the present invention. In contrast, Applicants refer the Examiners to FIG. 7 of the present invention which illustrates custom controls 602 that are used to edit properties of the object itself. Such custom ActiveX controls may take the form of a

combo-box (to select a color), a button (to click in a diagram to identify a center point), a slide bar to define a radius, etc. (see page 22, line 13-page 23, line 4). Thus, rather than showing properties of an ActiveX control (as in Ramchandani), the present invention uses ActiveX controls on a per property basis to edit properties of an object. Such a teaching is wholly and completely lacking from Ramchandani. Nowhere in Ramchandani is there any remote suggestion to use an ActiveX control that defines a user interface for editing a property of an object. Further, nowhere in Ramchandani is there any capability to use such an ActiveX control user interface in a property list (as both claimed and illustrated in FIG. 7).

Moreover, the various elements of Applicants' claimed invention together provide operational advantages over Ramchandani. In addition, Applicants' invention solves problems not recognized by Ramchandani.

Thus, Applicants submit that independent claims 1, 5, 9, 13, and 17 are allowable over Ramchandani. Further, dependent claims 2-4, 6-8, 10-12, 14-16, and 18 are submitted to be allowable over Ramchandani in the same manner, because they are dependent on independent claims 1, 5, 9, 13, and 17 respectively, and thus contain all the limitations of the independent claims. In addition, dependent claims 2-4, 6-8, 10-12, 14-16, and 18 recite additional novel elements not shown by Ramchandani.

V. Conclusion

In view of the above, it is submitted that this application is now in good order for allowance and such allowance is respectfully solicited. Should the Examiner believe minor matters still remain that can be resolved in a telephone interview, the Examiner is urged to call Applicants' undersigned attorney.

Respectfully submitted,

GATES & COOPER LLP
Attorneys for Applicant(s)

Howard Hughes Center
6701 Center Drive West, Suite 1050
Los Angeles, California 90045
(310) 641-8797

Date: May 18, 2007

By: Jason S. Feldmar/
Name: Jason S. Feldmar
Reg. No.: 39,187

JSF/kmk

G&C 30566.296-US-U1